

Examen de Entornos de Desarrollo

Comente brevemente las siguientes afirmaciones: (2 puntos)

- Dado un código breve y sencillo, realizado por un programador experimentado, podemos afirmar que no tendrá errores.

Como decía Edsger Dijkstra, por muy breve y sencillo que sea el código, es prácticamente imposible realizar pruebas exhaustivas a un programa ya que son muy costosas, y lo que se hace es llegar a un punto intermedio en el que no va a haber defectos importantes, o muchos defectos. Se busca que la aplicación tenga un funcionamiento aceptable. Muchas veces también se busca un rendimiento determinado y con eso nos basta. Las estrategias de las pruebas deben cubrir algunos aspectos que se definen al comienzo de la fase de pruebas.

- Es posible garantizar que un código no tiene errores, siempre que se hagan las pruebas unitarias por un equipo de pruebas especializado.

Igual que antes, nunca se garantiza que el código esté 100% libre de errores, además de que habitualmente, las pruebas no las hacen el equipo que desarrolla la aplicación, ya que saben lo que funciona y lo que no, y si supieran donde están los errores los corregirían, y por eso el equipo de pruebas suele ser un equipo exterior al de desarrollo.

Esto garantiza que al no conocer el código en profundidad, las pruebas serán más duras lo que aumentará la posibilidad de encontrar fallos.

Explique brevemente las estrategias de prueba de caja blanca para las siguientes estructuras: (2 puntos)

Para las pruebas de caja blanca se seguirán una serie de pasos, entre ellos por ejemplo pruebas de cobertura de código, y pruebas de condiciones y pruebas de bucles.

En las pruebas de cobertura de código se buscará que al menos se ejecute una vez cada línea del código si se puede, y esta prueba es común siempre.

- Selectiva.

Para las pruebas selectivas se utilizarán las pruebas de condiciones. Puede haber varias condiciones simples o compuestas, y se buscará probar al menos una vez cada condición que se cumpla, y otra vez que no se cumpla. De esta forma intentamos al menos probar todas las líneas de código posibles.

- Iterativa.

En las iterativas buscaremos hacer las pruebas de bucles en varias formas.

Repetir el máximo, el máximo +1 y el máximo -1 para ver si el resultado es esperado.

Repetir el bucle cero y una veces.

Repetir el bucle un número determinado de veces.

En los bucles anidados se buscará hacer las mismas pruebas para dichos bucles con la misma intención.

También daremos valores límite a los bucles para ver la reacción, como por ejemplo usando el cero, número fuera del rango, etc.

Diseñe el código y las pruebas de caja negra y blanca para una clase Calculadora, que tendrá los siguientes atributos y métodos: (6 puntos)

- **public string nombreCalculadora;**
- **public int[] números;**

- **public Calculadora(string nombre, int[] números):** constructor que inicia el nombre y los números que tendrá para los cálculos.
- **public void multiplicar(int x):** cada dato interno lo modifica multiplicándolo por x.
- **public int númeroRepeticiones(int número):** devuelve las veces que aparece el número dentro del vector. Si no aparece, produce una excepción.
- **Sobreescribir el método equals.** Dos calculadoras son iguales si tienen el mismo nombre y los mismos datos.